Podstawy Przetwarzania Danych

Wykład 8: Tworzenie środowiska testowego

dr inż. Marcin Luckner
mluckner@mini.pw.edu.pl

Wydział Matematyki i Nauk Informacyjnych

Wersja 1.1
5 marca 2021

Fundusze Europejskie Wiedza Edukacja Rozwój
Rzeczpospolita Polska
Politechnika Warszawska
Unia Europejska Europejski Fundusz Społeczny

Projekt „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca” współfinansowany jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zadanie 10 pn. „Modyfikacja programów studiów na kierunkach prowadzonych przez Wydział Matematyki i Nauk Informacyjnych”, realizowane w ramach projektu „NERW 2 PW. Nauka – Edukacja – Rozwój – Współpraca”, współfinansowanego jest ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Zbiór testowy

- Niech $\hat{d}(\mathbf{x}; \mathcal{L}_n)$ oznacza klasyfikator skonstruowany za pomocą próby uczącej $\mathcal{L}_n$.
- Załóżmy, że błąd klasyfikacji będziemy określać jako prawdopodobieństwo zaistnienia błędnej klasyfikacji.

$$\hat{e} \equiv e(\hat{d}) = \Pr\left(e(\mathbf{X}) \neq Y | \mathcal{L}_n\right).$$

- Jeżeli istnieje *niezależna* próba testowa $\mathcal{T}_m = \{(\mathbf{X}_1^t, Y_1^t), \ldots, (\mathbf{X}_m^t, Y_m^t)\}$ możemy estymować błąd klasyfikatora jako

$$\hat{e}_T = \frac{1}{m} \sum_{j=1}^{m} I\left(\hat{d}\left(\mathbf{X}_j^t; \mathcal{L}_n\right) \neq Y_j^t\right).$$

Alternatywne metody testowania

- Estymacja błędu na podstawie zbioru testowego $\mathcal{T}_m$ niezależnego od próbki uczącej $\mathcal{L}_n$ jest najlepszym możliwym podejściem.
- Jednakże czasami, ze względu na rozmiar danych lub ich specyficzny charakter, nie dysponujemy niezależnym zbiorem testowym.
- Alternatywne metody testowania [Krzyśko et al., 2008]
  - Metoda ponownego podstawienia.
  - Metoda sprawdzenia krzyżowego.
  - Metoda Jackknife.
  - Metoda prób bootstrapowych.

Metoda ponownego podstawiania

- Naturalną oceną poziomu błędu jest wartość estymatora ponownego podstawienia (redystrybucji) *resubstitution estimator*.
- Wyliczamy błąd dla zbioru uczącego, który służy też za zbiór testowy

$$\hat{e}_R = \frac{1}{n} \sum_{j=1}^{n} I\left(\hat{d}\left(\mathbf{X}_j; \mathcal{L}_n\right) \neq Y_j\right).$$

- Ponieważ próba ucząca jest też próbą testową estymator ten jest obciążony i zaniża rzeczywistą wartość błędu.
- Metoda może też prowadzić do przeuczenia klasyfikatora, czyli do utraty zdolności generalizacji reguł klasyfikacyjnych.

Podział próby

- Możemy podzielić próbę $\mathcal{L}_n$ na dwa podzbiory uczący i testowy.
- Metoda podziału próby (*holdout method*) powoduje jednak, że klasyfikator $\hat{d}$ będzie uczony tylko na części danych i może skutkować zaniżeniem estymacji błędu.
- Rozwiązaniem jest zastosowanie sprawdzenia krzyżowego (*cross-validation*), który stosuje wielokrotny podział próby do estymacji błędu.

Metoda sprawdzania krzyżowego

- Oznaczmy przez $\mathcal{L}_n^{(-j)}$ próbę uczącą $\mathcal{L}_n$ z której usunięto obserwację $\mathbf{Z}_j = (\mathbf{X}_j; Y_j)$.
- Klasyfikator tworzymy na zbiorze $\mathcal{L}_n^{(-j)}$ a następnie testuje na jednej obserwacji $\mathbf{Z}_j$.
  - Z powyższego powodu metodę nazywa się też *leave-one-out* LOO.
- Operację powtarza się n razy, dla każdej obserwacji $\mathbf{Z}_j$ z osobna.
- Estymator przybiera postać

$$\hat{e}_{CV} = \frac{1}{n} \sum_{j=1}^{n} I\left(\hat{d}\left(\mathbf{X}_j; \mathcal{L}_n^{(-j)}\right) \neq Y_j\right).$$

Ograniczenia sprawdzania krzyżowego

- Ze względu na testowanie budowanych klasyfikatorów na pojedynczej obserwacji, powstały estymator ma dużą wariancję.
- Ponadto wymaga konstrukcji n klasyfikatorów co może być zbyt kosztownym podejściem przy analizie większych zbiorów danych lub przy stosowaniu długotrwałych metod uczących.
- Rozwiązaniem problemu jest v-krokowa metoda metoda sprawdzania krzyżowego *v-fold cross-validation method*.

Metoda v-krokowego sprawdzania krzyżowego

- Próba ucząca $\mathcal{L}_n$ jest dzielona na v losowych próbek.
- $v-1$ próbek tworzy próbę uczącą, a jedna próbę testową.
- Operację budowy klasyfikatora powtarza się v razy.
- Estymator przybiera postać

$$\hat{e}_{vCV} = \frac{1}{n}\sum_{i=1}^{v}\sum_{j=1}^{n} I\left(\mathcal{Z}_j \in \tilde{\mathcal{L}}_n^{(i)}\right) I\left(\hat{d}\left(\mathbf{X}_j; \tilde{\mathcal{L}}_n^{(-i)}\right) \neq Y_j\right).$$

gdzie

- $\tilde{\mathcal{L}}_n^{(1)}, \ldots, \tilde{\mathcal{L}}_n^{(v)}$ jest losowym podzbiorem próby $\mathcal{L}_n$ na równoliczne zbiory.
- $\tilde{\mathcal{L}}_n^{(-i)} = \mathcal{L}_n \setminus \tilde{\mathcal{L}}_n^{(i)}$ dla $i = 1, 2, \ldots, v$

Omówienie v-krokowego sprawdzania krzyżowego

- Metoda v-krokowego sprawdzania krzyżowego daje mniejsze obciążenie błędu niż metoda podziału na próby i wymaga mniejszej liczby klasyfikatorów niż metoda sprawdzania krzyżowego o ile $v < n$.
- W zagadnieniu estymacji aktualnego poziomu błędu zalecane jest dobranie wartości $v = 10$ [Webb, 2003].
- W celu praktycznego wykorzystania oceny modeli wybieramy spośród stworzonych klasyfikatorów ten o najmniejszym błędzie.

Metoda prób bootstrapowych

- Próbą bootstrapową nazywamy próbę n elementową, pobraną z n elementowej próbki, za pomocą n krotnego losowania ze zwracaniem.item W tak zbudowanej próbie znajduje się około 63 procent obserwacji.
- Pozostałe dane mogą posłużyć jako zbiór testowy.
- Dodatkowo istnieje szereg estymatorów błędu dla metody prób bootstrapowych.

Estymator błędu prób bootstrapowych

- Dla ciągu B prób bootstrapowych $\mathcal{L}_n^{*1}, \mathcal{L}_n^{*2}, \ldots, \mathcal{L}_n^{*B}$ możemy wyliczyć błąd jako błąd ponownego podstawienia plus obciążenie elementów nieujętych w próbce

$$\hat{e}_{B_1} = \hat{e}_R + \frac{1}{Bn} \sum_{b=1}^{B} \sum_{j=1}^{n} I\left(\hat{d}\left(\mathbf{X}_j; \tilde{\mathcal{L}}_n^{*b} \neq Y_j\right)\right) - \frac{1}{Bn} \sum_{b=1}^{B} \sum_{j=1, \mathbf{Z}_j \in \mathcal{L}_n^{*b}}^{n} I\left(\hat{d}\left(\mathbf{X}_j; \tilde{\mathcal{L}}_n^{*b} \neq Y_j\right)\right)$$

- Możemy też ograniczyć się do elementów spoza próbki

$$\hat{e}_{B_2} = \frac{1}{B} \sum_{b=1}^{B} \frac{\sum_{j=1}^{n} I\left(\mathbf{Z}_j \notin \mathcal{L}_n^{*b}\right) I\left(\hat{d}\left(\mathbf{X}_j; \tilde{\mathcal{L}}_n^{*b} \neq Y_j\right)\right)}{\sum_{j=1}^{n} I\left(\mathbf{Z}_j \notin \mathcal{L}_n^{*b}\right)}$$

Modyfikacje estymatorów

- Na podstawie poprzednich estymatorów, które stosowały metodę sprawdzania krzyżowego dla prób bootstrapowych, możemy wyprowadzić nowe estymatory.
- Pamiętając, że z prawdopodobieństwem około 37% nie wylosujemy danej obserwacji do próbki stosujemy

$$\hat{e}_{.632} = .368\hat{e}_R + .632\hat{e}_{B_2}$$

- Alternatywnie

$$\hat{e}_{.632+} = (1 - \omega)\hat{e}_R + \omega\hat{e}_{B_2}$$

Parametry estymatora $\hat{e}_{.632+}$

- Estymator $\hat{e}_{.632+}$ jest parametryzowany wyrażeniem ω

$$\omega = \frac{0.632}{1 - 0.368R},$$

- gdzie

$$R = \begin{cases} 1, \text{ jeżeli } \gamma \leqslant \hat{e}_{B_2}, \\ \frac{\hat{e}_{B_2} - \hat{e}_R}{\gamma - \hat{e}_R}, \text{ jeżeli } \gamma, \hat{e}_{B_2} > \hat{e}_R, \\ 0, \text{ w p.p.} \end{cases}$$

- dla parametru

$$\gamma = \frac{1}{n^2} \sum_{j=1}^{n} \sum_{k=1}^{n} I\left(\hat{d}\left(\mathbf{x}_j; \tilde{\mathcal{L}}_n \neq Y_j\right)\right)$$

Jackknife

- Jackknife jest techniką próbkowania , wcześniejszą niż metoda prób bootstrapowych, pozwalającą na estymację obciążenia [Washington, 2005].
- Estymacja parametru odbywa się poprzez systematyczne usuwanie obserwacji ze zboru i obliczanie estymacji, a następnie uśrednienie wyników.
- Dla n elementowej próbki, estymacja jackknife jest wyliczana na podstawie uśrednienia estymacji dla każdej próbki o rozmiarze $n - 1$.

Estymator Jackknife

- Estymator parametru metodą jackknife można uzyskać poprzez estymację parametru dla każdej próbki powstałej z usunięcia i-tej obserwacji.
- Chcąc oszacować średnią wartość zmiennej x wyliczamy średnią dla każdej próbki bez i-tej obserwacji

$$\bar{x}_i = \frac{1}{n-1} \sum_{j=1, j\neq i}^{n} x_j, \qquad i = 1, \ldots, n.$$

- Następnie uśredniamy uzyskane estymatory, aby uzyskać estymację dla całego zbioru

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} \bar{x}_i$$

Estymacja błędu metodą Jackknife

- Technika jackknife może być użyta do estymacji obciążenia estymatora wyliczanego dla całej próbki.
- Możemy wyliczyć

$$\hat{e}_{(.)} = \frac{1}{n}\sum_{i=1}^{n}\hat{e}_{(i)}$$

gdzie $\hat{e}_{(i)}$ jest błędem wyliczonym dla zbioru $\mathcal{L}_n^{(-i)}$ jak w metodzie LOO, a $\hat{e}_{(.)}$ jest estymatorem uśredniającym te szacowania.

- Obciążenie estymatora dla całej próbki $\hat{e}$ wyliczamy jako

$$\widehat{\text{bias}}_{(\text{e})} = (n-1)(\hat{e}_{(.)} - \hat{e})$$

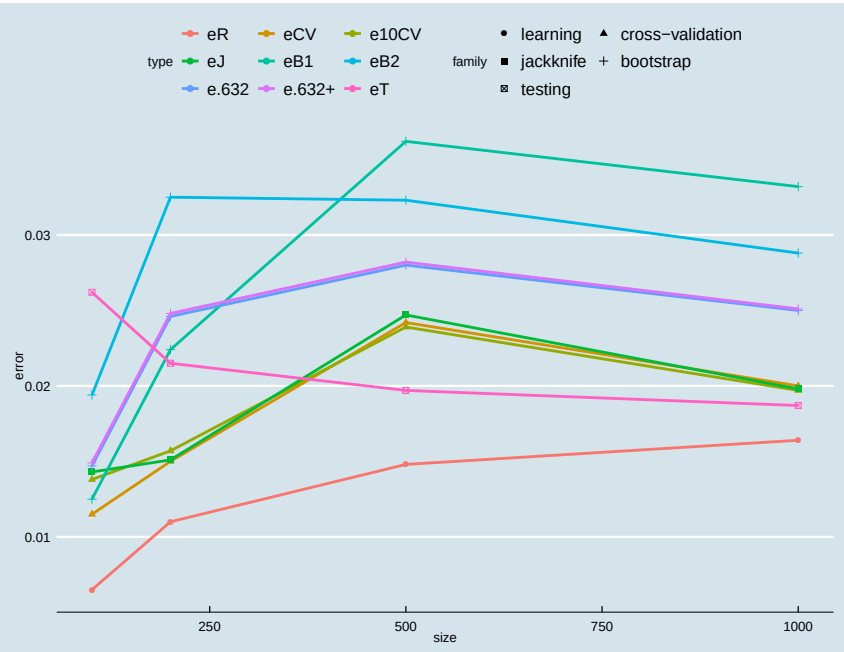
- To pozwala estymować błąd jako

$$\hat{e}_{\text{jack}} = \hat{e} - \widehat{\text{bias}}_{(\text{e})} = n\hat{e} - (n-1)\hat{e}_{(.)}$$

Porównanie estymatorów

- Przeprowadzono porównanie metod estymacji błędu na wygenerowanych danych [Krzyśko et al., 2008].
- Dla dwóch klas wygenerowano po 100, 200, 500 i 1000 przykładów.
- Wyniki działania estymatorów porównano z wynikiem na próbie testowej składającej się z 100 tysięcy elementów.
- Wyniki uzyskane na zbiorze testowym potraktujemy jako wyniki referencyjne.

Wyniki porównania estymatorów



Rysunek 1: Porównanie estymatorów

- Dla próbki 100 elementowej wszystkie estymatory zaniżają wartość błędu (próbka jest zbyt mała).
- Estymator ponownego podstawienia zawsze powoduje niedoszacowanie.
- Metody bootstrapowe dają wyższe oszacowania niż metody sprawdzenia krzyżowego.

Ograniczenia estymatorów

- Metody sprawdzania krzyżowego są szeroko stosowane do porównywania klasyfikatorów. Jednakże należy pamiętać, że opierają się one na doborze podzbiorów zbioru uczącego (podobnie jak metody bootstrapowe).
- Z tego też powodu nie należy ich używać w sytuacji, gdy klasyfikator będzie używany na wyraźnie innych danych niż dane uczące.
- Dotyczy to w szczególności analizy danych zmiennych w czasie. W takim wypadku powinniśmy wyodrębniać zbiór testowy.

Wzmacnianie klasyfikatorów

- Próbkowanie stosowane przez bootstrap może zostać wykorzystane do wzmacniania klasyfikatorów.
- Algorytm bagging jest metodą redukcji zmienności (wariancji) klasyfikatorów).
- Algorytm boosting redukuje wariancję i obciążenie klasyfikatorów.
- Metody są stosowane do wzmacnianie klasyfikatorów o niskiej stabilności.
  - drzewa decyzyjne, sieci neuronowe.

Algorytm bagging

1. Dla $b = 1, 2, \ldots, B$:
   1.1 Pobierz próbę bootstrapową $\mathcal{L}_n^{*b}$ z próby $\mathcal{L}_n$.
   1.2 Konstruuj klasyfikator $\hat{d}_b$ dla próby bootstrapowej $\mathcal{L}_n^{*b}$
2. Klasyfikuj obserwację $\mathbf{x}$ wg reguły

$$\hat{d}_{Bag1}(x) = \arg\max_{1 \leqslant k \leqslant K} \sum_{b=1}^{B} I\left(\hat{d}_b(x) = k\right).$$

Uwagi do algorytmu bagging

- Metoda wybiera końcowy wynik na drodze głosowania.
- Ponieważ około 37 procent elementów nie było użytych do budowy klasyfikatorów to mogą one posłużyć jako zbiór walidacyjny.
- Wyniki uzyskane na zbiorze walidacyjnym służą do budowy ważonej wersji algorytmu w którym lepsze klasyfikatory mają wyższą wagę głosów

Ważony algorytm bagging

1. Dla $b = 1, 2, \ldots, B$:
   - 1.1 Pobierz próbę bootstrapową $\mathcal{L}_n^{*b}$ z próby $\mathcal{L}_n$.
   - 1.2 Konstruuj klasyfikator $\hat{d}_b$ dla próby bootstrapowej $\mathcal{L}_n^{*b}$
   - 1.3 Oblicz
$$w^{(b)} = \frac{\sum_{j=1}^{n} [I(z_j \notin \mathcal{L}_n^{*b}) I(\hat{d}_b = y_j)]}{\sum_{j=1}^{n} I(z_j \notin \mathcal{L}_n^{*b})}$$
2. Normalizuj $w^{(b)} = w^{(b)} / \sum_{k=1}^{B} w^{(k)}$
3. Klasyfikuj obserwację $\mathbf{x}$ wg reguły

$$\hat{d}_{Bag1}(x) = \arg\max_{1 \leqslant k \leqslant K} \sum_{b=1}^{B} w^{(b)} I\left(\hat{d}_b(x) = k\right).$$

Idea algorytmu boosting

- Algorytm bagging losuje elementy do zbioru uczącego z rozkładu jednostajnego.
- Algorytm boosting proponuje modyfikować dynamicznie wagi obserwacji tak aby zwiększać prawdopodobieństwo wybrania źle rozpoznane elementów do zbioru uczącego.
- Algorytm wprowadza też metodę oceny poszczególnych klasyfikatorów stosowaną przy wyliczaniu końcowej klasyfikacji.
- Klasyczną implementacją algorytmu boostingu jest algorytm AdaBoost.

Algorytm AdaBoost I

1. Ustal wektor wag $\mathbf{w}^{(1)} w_j^{(1)} = 1/n, j = 1, 2, \ldots, n$.
2. Dla $b = 1, 2, \ldots, B$:
   - 2.1 Pobierz próbę bootstrapową $\mathcal{L}_n^{*b}$ z próby $\mathcal{L}_n$ według rozkładu $\Pr(z_j \in \mathcal{L}_n^{*b}) = w_j^{(b)} \wedge j = 1, 2, \ldots, n$.
   - 2.2 Konstruuj klasyfikator $\hat{d}_b$ dla próby bootstrapowej $\mathcal{L}_n^{*b}$
   - 2.3 Oblicz $\hat{e}_b = \sum_{j=1}^{n} w_j^{(b)} l_j^{(b)}$ gdzie

$$l_j^{(b)} = \begin{cases} 1, & \text{jeżeli } \hat{d}_b(x) \neq y_j, \\ 0, & \text{w p.p.} \end{cases}$$

   - 2.4 Jeżeli $\hat{e}_b \in (0, 0.5)$, oblicz $\beta_b = \frac{\hat{e}_b}{1-\hat{e}_b}$. W przeciwnym wypadku $w_j^{(1)} = 1/n, j = 1, 2, \ldots, n$ i krok 2.1.

Algorytm AdaBoost II

2.5 Aktualizuj wagi

$$w_j^{(b+1)} = \frac{w_j^{(b)} \beta_b^{1-l_j^{(b)}}}{\sum_{i=1}^{n} w_i^{(b)} \beta_b^{1-l_i^{(b)}}}, j = 1, 2, \ldots, n$$

3. Klasyfikuj obserwację **x** wg reguły

$$\hat{d}_{AdaB}(x) = \arg\max_{1 \leqslant k \leqslant K} \sum_{b=1}^{B} \left[\ln\left(\frac{1}{\beta_b}\right) I\left(\hat{d}_b(x) = k\right) \right].$$

Uwagi do algorytmu Adaboost

- Algorytm AdaBoost oryginalnie został stworzony do klasyfikacji dwóch klas.
- Istnieją jednak algorytmy boostingu dedykowane do różnych zadań.
- Między innymi:
  - Klasyfikacja wieloklasowa,
    - XGBoost [Chen and Guestrin, 2016].
  - Regresja.
    - MART [Friedman, 2000].
  - Zbiory niezrównoważone.
    - RUSBoost [Seiffert et al., 2010].

Porównanie algorytmów wzmacniania

- W zadaniu lokalizacji telefonu komórkowego na podstawie siły sygnału sieci GSM porównano różne algorytmy wzmacniania [Górak et al., 2016].
- Testy były przeprowadzane w wielopiętrowym budynku.
- Oddzielnie rozważano błąd poprawnego rozpoznania piętra i błąd położenia liczony w płaszczyźnie poziomej.

Wyniki lokalizacji

Method	Accuracy [%]	Average error [floor]
AdaBoost	33	1.50
Bagging Classification	56	0.83
Bagging Regression	43	0.81
Least Squares Boosting	29	1.07

Rysunek 2: Rozpoznanie piętra [Górak et al., 2016]

	Mean	Median	80th percentile
LS Boosting	10.22	9.15	14.75
Bagging	8.15	6.76	12.16

Rysunek 3: Lokalizacja pozioma, błąd w metrach [Górak et al., 2016]

Bibliografia I

[Chen and Guestrin, 2016] Chen, T. and Guestrin, C. (2016).

Xgboost: A scalable tree boosting system.

In Krishnapuram, B., Shah, M., Smola, A. J., Aggarwal, C., Shen, D., and Rastogi, R., editors, *KDD*, pages 785–794. ACM.

[Friedman, 2000] Friedman, J. H. (2000).

Greedy function approximation: A gradient boosting machine.

Annals of Statistics, 29:1189–1232.

[Górak et al., 2016] Górak, R., Luckner, M., Okulewicz, M., Porter-Sobieraj, J., and Wawrzyniak, P. (2016).

Indoor localisation based on GSM signals: Multistorey building study.

Mobile Information Systems, 2016:2719576:1–2719576:17.

Bibliografia II

[Krzyśko et al., 2008] Krzyśko, M., Wołyński, W., Górecki, T., and Skorzybut, M. (2008).

Systemy uczące się. Rozpoznawanie wzorców analiza skupień i redukcja wymiarowości.

WNT.

[Seiffert et al., 2010] Seiffert, C., Khoshgoftaar, T. M., Hulse, J. V., and Napolitano, A. (2010).

Rusboost: A hybrid approach to alleviating class imbalance.

IEEE Trans. Systems, Man, and Cybernetics, Part A, 40(1):185–197.

[Washington, 2005] Washington, S. S. (2005).

Resampling Data : Using a Statistical Jackknife.

Technical Report 1.

[Webb, 2003] Webb, A. R. (2003).

Statistical Pattern Recognition.

John Wiley & Sons.